**Lab 7 – Servos**

***The lab introduces shows how the microcontroller will control a standard hobby servo. You can find servos in most radio controlled toys. On a larger scale, they are called actuators. In this lab, you will experiment with servo movement.***

**Hardware Required**Arduino Board servo & miscellaneous jumper wires (picture of servo shown below)

**Circuit**

The servo has three wires extending from its base. Connect the black wire to ground, the red wire to pos 5+ volts, and the signal wire to pin 6. Notice on the board that pin 6 says PWM next to it. That stands for Pulse Width Modulation (PWM).

Web search and write a definition of PWM :

Identify and list the other Arduino pins with the letters PWM next to them :

**Basic Code Explanation**

In the program below the servo sweeps from one direction to another.

**Part 1 - Directions**

Run sample code 1.

Sample code 1

------------------------------------------------------------------

#include <Servo.h> //make sure to capitalize the ‘S’ in Servo.h  
  
Servo myservo;   // create servo object to control a servo  
 // this servo’s name is myservo   
int angle = 0;   // variable to store the servo position  
void setup()  
{  
  myservo.attach(6);  // attaches the servo on pin 6 to the servo object  
}  
void loop()  
{  
  for(angle = 0; angle < 180; angle += 1)  // goes from 0 degrees to 180 degrees  
  {                                        // in steps of 1 degree  
    myservo.write(angle);    // tell servo to go to position in variable 'angle'  
    delay(30);                       // waits 20ms between servo commands  
  }  
  for(angle = 180; angle >= 1; angle -= 1) // goes from 180 degrees to 0 degrees  
  {  
    myservo.write(angle);     // tell servo to go to position in variable 'pos'  
    delay(30);                       // waits 20ms between servo commands  
  }  
}

------------------------------------------------------------------

Once the program works, observe its behavior. Obtain the instructor’s initials here (\_\_\_)

**Modify the Program**Change the delay to 10ms and observe the servo’s behavior

Change the delay to 50ms and observe the servo’s behavior (\_\_\_)

Describe the difference between the three (including the initial program) programs?

**Part 2 - Directions**

Leave the servo connections alone. Connect the temperature resistor to Pin A0 (analog pin zero) with two 10kOhm resistors in parallel with it. Also connect an external LED to pin 13. The LED will simulate the two positions of your servo. You will simulate a temperature controlled damper with the code Run sample code 2

Sample code 2

------------------------------------------------------------------

/\*

This code moves a servo to two positions (fixed values "pos0" and "pos1") based on the analog input value. It will display the value on the serial monitor.

These two positions are set values. The two 10kOhm resistors (in parallel with the temperature resistor) provide a value near 324 at room temperature.

\*/

#include <Servo.h>

Servo myservo; // create servo object to control a servo

int pos0 = 40; // variable to store the servo position (closed)

int pos1 = 160; // variable to store the servo position (open)

int ledPin = 13; // led connected to pin13

void setup () {

Serial.begin(9600);

myservo.attach(6); // Attaches the servo on pin9 to the servo object.

pinMode(ledPin, OUTPUT); // Assign pin to output.

} // You may have to change the 350 to another

// number that works for your program.

void loop () {

int temp = analogRead(0);

delay(3000); // Time between cycles

Serial.print(temp); // Print the value to serial monitor

if(temp > 350) { // Greater than 350 equals hot condition

myservo.write(pos0); // Move servo to position in variable “pos”

digitalWrite(ledPin, HIGH); // Turn LED on

}

else if (temp < 350) { // Less than 350 equals cold temp

myservo.write(pos1); // Move servo to position in variable “pos2”

digitalWrite(ledPin, LOW); // Turn LED off

}

}

------------------------------------------------------------------

Once the program works, observe its behavior. Obtain the instructor’s initials here (\_\_\_)

**Modify the Program**Change the delay time to a greater number and watch how it affects the program’s response.

Change the delay time to a smaller number and watch how it affects the program’s response.

Summarize your findings

List four places where you could use something like this.

1

2

3

4

**Modify the Circuit**Disconnect power and swap the temperature resistor with the photocell. Now use the light values to control the servo. You will need to adjust the analog values to match your program. When finished, obtain instructor’s signature (\_\_\_).

List five places where you could use something like this.

1

2

3

4

5